



SharkFest '18 Europe



BGP is not only a TCP session

Learning about the
protocol that holds
networks together

<https://goo.gl/mh3ex4>

Werner Fischer

Principal Consultant | avodaq AG



Agenda



- History and RFCs
- Direction for further research
- BGP Notifications
- Authentication and Security
- 2 versus 4 Bytes
- Communities
- BGP-LS
- BGP Additional Path
- BGP EVPN
- BGP Graceful Restart
- Funny things with BGP
- Wrap-up





About me



- From Germany
- More than a decade Dual-CCIE (R/S, Security)
- Sniffer Certified Master
- Wireshark Certified Network Analyst
- Dual VMware Certified Professional (VCP-DCV, VCP-NX)
- IPv6 Forum Certified Engineer (Gold)
- Round about 20 years in the networking area like Wireshark/Ethereal





History and RFCs





History of BGP



BGP
Boundary
Gateway
Protocol

block format
block length
version number
block type
holddown timer

2 bytes
1 byte
2 bytes (seconds)
2 bytes (minutes)

version is currently 1

types:
open - 1
update - 2
notification - 4
keepalive - 8

open:
my AS # 2 byte
link type 1 byte

up - 1
down - 2
internal - 4
H-link - 8 (not used in update distribute field)

auth type code 1 byte
0 - none

authentication variable

update:
network # 4 bytes
first hop gateway 4 bytes
metric 2 bytes
count of AS 1 byte
{ direction 1 byte }
{ AS # 2 byte } repeat "count" times

notification:
open code 2 bytes
data variable

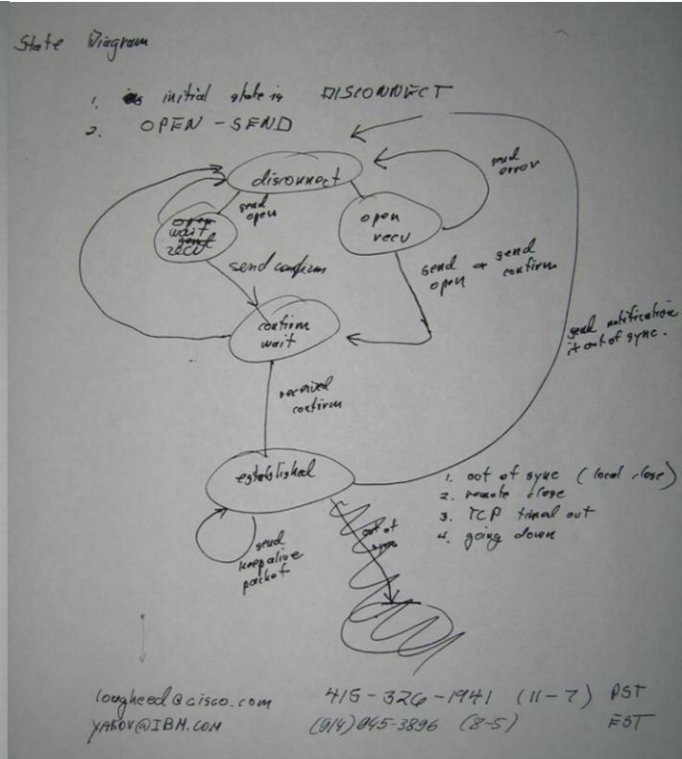
operations

- link type error in open
- my view of correct link type (1 byte)
- unknown auth type code
- no data
- authentication failure (no data)
- update error - data is block in error
~~routing loop in update~~
~~two phase error in update~~
data is subcode (2 byte) followed by
update block in question (1 network only)

subcodes -

- invalid network field
- invalid first hop gw
- invalid direction code
- invalid AS
- routing loop
- two-phase error

- connection out of sync - data is last block received (TCP close after packet sent)
- open continued
- invalid block type (data is 1 byte block type)
- invalid version number (data is 1 byte version)





BGP (Border Gateway Protocol)



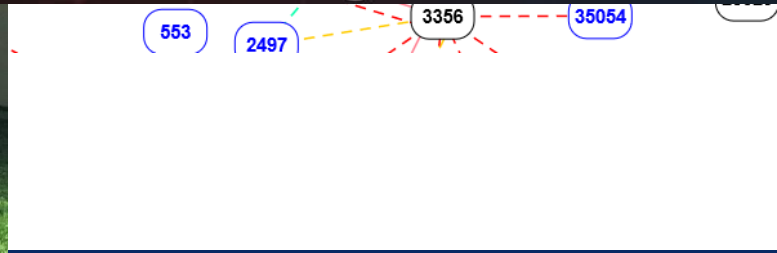
- BGP is a standardized EGP designed to exchange routing and reachability information between autonomous systems (ASs)
- “Is less chatty than its link-state siblings”
- “Does not require routing state to be periodically refreshed, unlike OSPF or IS-IS”
- Many stable vendor implementations
- BGP is a multi-protocol routing engine, capable of announcing different prefixes (e.g. IPv4 and IPv6 and others)



BGP, Security and Crypto Currency



BGP Hijack of Amazon DNS to Steal Crypto Currency

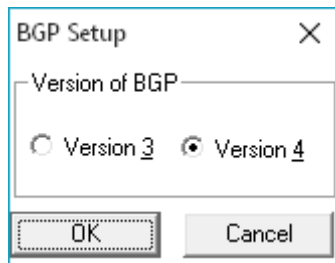




Former versions of BGP – hint in RFC 1771



Note that quite often BGP, as specified in RFC 1105, is referred to as BGP-1, BGP, as specified in RFC 1163, is referred to as BGP-2, BGP, as specified in RFC1267 is referred to as BGP-3, and BGP, as specified in this document is referred to as BGP-4.





BGP-4 – Basic RFCs



- RFC 1771 - A Border Gateway Protocol 4 (BGP-4)
- RFC 1863 - A BGP/IDRP Route Server alternative to a full mesh routing
- RFC 1997 - BGP Communities Attribute
- **RFC 2385 - Protection of BGP Sessions via the TCP MD5 Signature Option**
- RFC 2545 - Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
- **RFC 4271 - A Border Gateway Protocol 4 (BGP-4)**
- **RFC 4760 - Multiprotocol Extensions for BGP-4**
- **RFC 5492 - Capabilities Advertisement with BGP-4**



BGP-4 – Advanced RFCs



- First - too much for 90 minutes! I personally need more than 90000 minutes to read and follow them all ;-)
- ...
- RFC 4360 - Route Target extended community
- RFC 4364 - BGP/MPLS IP Virtual Private Networks (VPNs)
- RFC 4384 - BGP Communities for Data Collection
- RFC 4724 - Graceful Restart Mechanism for BGP
- RFC 4761 - Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling
- RFC 5512 - The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute
- ...





BGP-4 – Advanced RFCs



- ...
- RFC 5549 - Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop
- RFC 5575 - Dissemination of Flow Specification Rules
- RFC 5668 - 4-Octet AS Specific BGP Extended Community
- RFC 5701 - IPv6 Address Specific BGP Extended Community Attribute
- RFC 5925 - The TCP Authentication Option
- RFC 6514 - BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs
- ...





BGP-4 – Advanced RFCs



- ...
- RFC 6515 - IPv4 and IPv6 Infrastructure Addresses in BGP Updates for Multicast VPN
- RFC 6793 - BGP Support for Four-Octet Autonomous System (AS) Number Space
- RFC 6811 - BGP Prefix Origin Validation
- RFC 6996 - Autonomous System (AS) Reservation for Private Use
- RFC 7153 - IANA Registries for BGP Extended Communities
- RFC 7300 - Reservation of Last Autonomous System (AS) Numbers
- RFC 7311 - The Accumulated IGP Metric Attribute for BGP
- RFC 7313 - Enhanced Route Refresh Capability for BGP-4
- ...





BGP-4 – Advanced RFCs



- ...
- RFC 7432 - BGP MPLS-Based Ethernet VPN
- RFC 7454 - BGP Operations and Security
- RFC 7543 - Covering Prefixes Outbound Route Filter for BGP-4
- RFC 7606 - Revised Error Handling for BGP UPDATE Messages
- RFC 7674 - Clarification of the Flowspec Redirect Extended Community
- RFC 7752 - North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP
- RFC 7900 - Extranet Multicast in BGP/IP MPLS VPNs
- RFC 7911 - Advertisement of Multiple Paths in BGP
- ...





BGP-4 – Advanced RFCs



- RFC 7938 - Use of BGP for Routing in Large-Scale Data Centers
- RFC 7999 - BLACKHOLE Community
- RFC 8092 - BGP Large Communities Attribute
- RFC 8097 - BGP Prefix Origin Validation State Extended Community
- RFC 8214 - Virtual Private Wire Service Support in Ethernet VPN
- RFC 8277 - Using BGP to Bind MPLS Labels to Address Prefixes
- RFC 8317 - Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) and Provider Backbone Bridging EVPN (PBB-EVPN)
- RFC 8326 - Graceful BGP Session Shutdown
- RFC 8365 - A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)
- RFC 8388 - Usage and Applicability of BGP MPLS-Based Ethernet VPN





Direction for further research





Searching for bugs – or for sample capture files 😊

Summary:

contains all of the strings

bgp

Search

Product:

Web sites
Wireshark

Component:

Build process
Capture file support (libwiretap)
Common utilities (libwsutil)
Dissection engine (libwireshark)
Documentation
Extras

Status:

UNCONFIRMED
CONFIRMED
IN_PROGRESS
INCOMPLETE
RESOLVED
VERIFIED

Resolution:

FIXED
NOTABUG
NOTOURBUG
LATER
REMIND

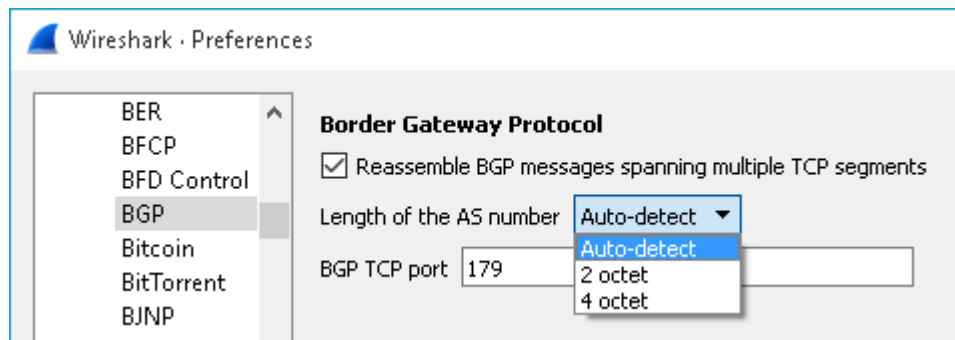




Wireshark Preferences



For BGP there are not so much options – really – you think so?



This AS number length topic will be also presented – stay tuned!



Field Name copy for DFE



- Copy the Field Name for DFE (Display Filter Expression)

The screenshot shows the Wireshark interface with a context menu open over a BGP UPDATE message field. The menu options are:

- Copy
- Show Packet Bytes... (Ctrl+Shift+O)
- Export Packet Bytes... (Ctrl+Shift+X)
- Wiki Protocol Page
- Filter Field Reference
- Protocol Preferences
- Decode As...
- Go to Linked Packet
- Show Linked Packet in New Window

The 'Field Name' option is highlighted with a red box. The keyboard shortcuts for the other options are:

- All Visible Items: Ctrl+Alt+Shift+A
- All Visible Selected Tree Items
- Description: Ctrl+Alt+Shift+D
- Field Name: Ctrl+Alt+Shift+F
- Value: Ctrl+Alt+Shift+V
- As Filter: Ctrl+Shift+C
- Copy Bytes as Hex + ASCII Dump
- ...as Hex Dump
- ...as Printable Text
- ...as a Hex Stream
- ...as Raw Binary
- ...as Escaped String

The background shows the packet list and packet bytes pane. The selected field is '0... .. = Optional: Not set' under the 'Flags' section of the 'Path Attribute - AS_PATH' field.



RFC 6996 - Autonomous System (AS) Reservation for Private Use



IANA has reserved, for Private Use, a contiguous block of 1023 Autonomous System numbers from the "16-bit Autonomous System Numbers" registry, namely 64512 - 65534 inclusive.

IANA has also reserved, for Private Use, a contiguous block of 94,967,295 Autonomous System numbers from the "32-bit Autonomous System Numbers" registry, namely 4200000000 - 4294967294 inclusive.

- <https://www.iana.org/assignments/as-numbers/as-numbers.xhtml>





BGP Notifications





BGP Errors via Notification - DFE



Wireshark - Display Filter Expression

Field Name	Relation
BGP - Border Gateway Protocol	is present
bgp.notify.major_error - Major error Code	==

Message Header Error
OPEN Message Error
UPDATE Message Error
Hold Timer Expired
Finite State Machine Error
Cease
CAPABILITY Message Error

Search: bgp.notify.major_error

bgp.notify.major_error == 1

Click OK to insert this filter

OK Cancel Help

Wireshark - Display Filter Expression

Field Name	Relation
BGP - Border Gateway Protocol	is present
bgp.notify.minor_error - Minor error Code (Message Header)	==
bgp.notify.minor_error.unknown - Unknown notification error	!=

Maximum Number of Prefixes Reached
Administratively Shutdown
Peer De-configured
Administratively Reset
Connection Rejected
Other Configuration Change
Connection Collision Resolution
Out of Resources

Search: bgp.notify.minor_error

bgp.notify.minor_error_cease == 1

Click OK to insert this filter

OK Cancel Help



BGP NOTIFICATIONs



Different Types of NOTIFICATIONs

▼ Border Gateway Protocol - NOTIFICATION Message

Marker: ffffffffffffffffffffffffffffffff

Length: 21

Type: NOTIFICATION Message (3)

Major error Code: Hold Timer Expired (4)

Minor error Code (Hold Timer Expired): 0

▼ Border Gateway Protocol - NOTIFICATION Message

Marker: ffffffffffffffffffffffffffffffff

Length: 21

Type: NOTIFICATION Message (3)

Major error Code: Cease (6)

Minor error Code (Cease): Unknown (0)

▼ Border Gateway Protocol - NOTIFICATION Message

Marker: ffffffffffffffffffffffffffffffff

Length: 21

Type: NOTIFICATION Message (3)

Major error Code: Cease (6)

Minor error Code (Cease): Administratively Reset (4)

▼ Border Gateway Protocol - NOTIFICATION Message

Marker: ffffffffffffffffffffffffffffffff

Length: 21

Type: NOTIFICATION Message (3)

Major error Code: Cease (6)

Minor error Code (Cease): Other Configuration Change (6)

→ DFE

bgp.notify.major_error

bgp.notify.minor_error



BGP NOTIFICATION with Data



NOTIFICATION

```
> Frame 1100: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 0
> Ethernet II, Src: PcsCompu_a3:d0:38 (08:00:27:a3:d0:38), Dst: DellEmc_ff:01:02 (00:01:44:ff:01:02)
> Internet Protocol Version 4, Src: 10.25.2.7 (10.25.2.7), Dst: 10.25.2.9 (10.25.2.9)
> Transmission Control Protocol, Src Port: 179, Dst Port: 44084, Seq: 635, Ack: 100, Len: 39
> Border Gateway Protocol - NOTIFICATION Message
  Marker: ffffffffffffffffffffffffffffffff
  Length: 39
  Type: NOTIFICATION Message (3)
  Major error Code: Cease (6)
  Minor error Code (Cease): Peer De-configured (3)
  Data: 506565722044652d636f6e666696775726564
```

```
0000 00 01 44 ff 01 02 08 00 27 a3 d0 38 08 00 45 00  .D.....'8..E
0010 00 5b 78 14 40 00 40 06 aa 47 0a 19 02 07 0a 19  [x.@. @.G.....
0020 02 09 00 b3 ac 34 1b f2 da f4 8e da d7 b4 80 18  ....4.....
0030 00 1d 2c d4 00 00 01 01 08 0a 24 0d e8 52 41 ee  ,.....$..RA..
0040 75 31 ff ff ff ff ff ff ff ff ff ff ff ff ff ff  u1.....
0050 ff ff 00 27 03 06 03 50 65 65 72 20 44 65 2d 63  ....P eer De-c
0060 6f 6e 66 69 67 75 72 65 64  onfigure d
```

Data:

This variable-length field is used to diagnose the reason for the NOTIFICATION. The contents of the Data field depend upon the Error Code and Error Subcode. See [Section 6](#) for more details.

Note that the length of the Data field can be determined from the message Length field by the formula:

$$\text{Message Length} = 21 + \text{Data Length}$$

The minimum length of the NOTIFICATION message is 21 octets (including message header).

Unless specified explicitly, the Data field of the NOTIFICATION message that is sent to indicate an error is empty.

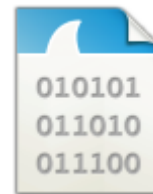
→ RFC 4271



NOTIFICATION

- ▼ Border Gateway Protocol - NOTIFICATION Message
 - Marker: ffffffffffffffffffffffffffffffffff
 - Length: 27
 - Type: NOTIFICATION Message (3)
 - Major error Code: OPEN Message Error (2)
 - Minor error Code (Open Message): Unsupported Capability (7)
- ▼ Capability: Multiprotocol extensions capability
 - Type: Multiprotocol extensions capability (1)
 - Length: 4
 - AFI: IPv6 (2)
 - Reserved: 00
 - SAFI: Unicast (1)

[henetv6-bgp-tunnel-packetcapture.cap](#)





Authentication and Security





TCP MD5 Authentication [RFC 2385]



BGP use the TCP option

Wireshark · Display Filter Expression

Field Name	Relation
▼ TCP · Transmission Control Protocol	is present
tcp.option_kind · Kind	==
	!=
	>
	<
	>=
	<=
	in

Value (Unsigned integer, 1 byte)
19

Predefined Values

- MD5 Signature Option
- SCPS Capabilities
- Selective Negative Acknowledgements
- Record Boundaries

Range (offset:length)

Search: tcp.option_kind

tcp.option_kind == 19

Click OK to insert this filter

OK Cancel Help

```
Transmission Control Protocol, Src Port: 179, Dst Port: 25820, Seq: 1, Ack: 46, Len: 45
  Source Port: 179
  Destination Port: 25820
  [Stream index: 0]
  [TCP Segment Len: 45]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 46 (relative sequence number)]
  Acknowledgment number: 46 (relative ack number)
  1010 ... = Header Length: 40 bytes (10)
  Flags: 0x018 (PSH, ACK)
  Window size value: 16339
  [Calculated window size: 16339]
  [Window size scaling factor: -2 (no window scaling used)]
  Checksum: 0x45e6 [unverified]
  [Checksum Status: Unverified]
  Urgent pointer: 0
  Options: (20 bytes), TCP MD5 signature, End of Option List (EOL)
    TCP Option - TCP MD5 signature
      Kind: MD5 Signature Option (19)
      Length: 18
      MD5 digest: 54fad66bd53fd9475a447025bb9c9b21
    TCP Option - End of Option List (EOL)
  [SEQ/ACK analysis]
  [Timestamps]
    [Time since first frame in this TCP stream: 0.015947000 seconds]
    [Time since previous frame in this TCP stream: 0.003958000 seconds]
  TCP payload (45 bytes)
  Border Gateway Protocol - OPEN Message
```




Mixing MD5 and TCP-AO



Draft was found for that

12.4. Backwards Compatibility

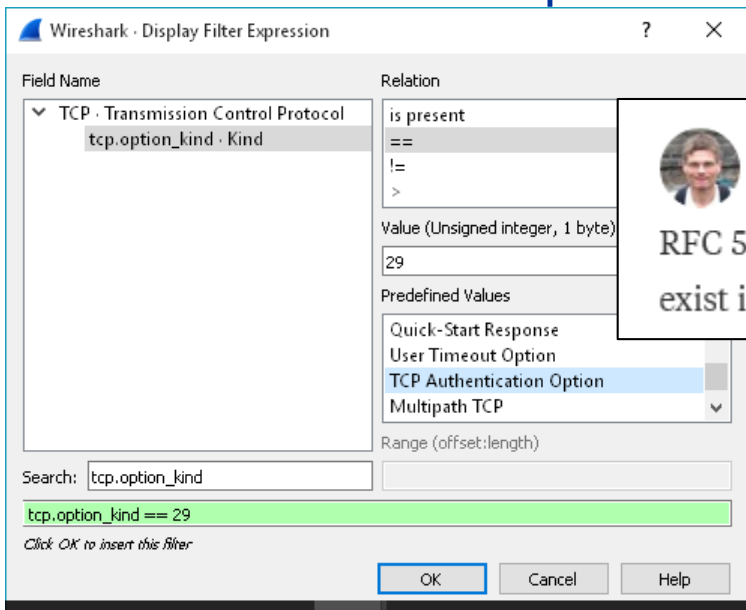
On any particular TCP connection, use of the TCP Enhanced Authentication Option precludes use of the TCP MD5 Signature Option. However, use of the TCP Enhanced Authentication Option on one connection does not preclude the use of the TCP MD5 Signature Option on another connection by the same system.



TCP-AO [RFC 5925]



BGP use the TCP option



The image shows the 'Wireshark · Display Filter Expression' dialog box. The 'Field Name' list is expanded to 'TCP · Transmission Control Protocol' and then to 'tcp.option_kind · Kind'. The 'Relation' dropdown is set to '=='. The 'Value (Unsigned integer, 1 byte)' field contains '29'. The 'Predefined Values' list is open, showing 'Quick-Start Response', 'User Timeout Option', 'TCP Authentication Option' (highlighted), and 'Multipath TCP'. The 'Search:' field contains 'tcp.option_kind'. The filter expression 'tcp.option_kind == 29' is displayed in a green box at the bottom. Below the filter expression, it says 'Click OK to insert this filter'. There are 'OK', 'Cancel', and 'Help' buttons at the bottom right.



Jakob Heitz, Principal Engineer at Cisco, developing BGP

Answered Apr 3 2017 · Author has **130** answers and **138.5k** answer views

RFC 5925 (TCP-AO) obsoletes RFC 2385 (MD5), but TCP-AO does not actually exist in any significant implementations, because of lack of interest.



Routing Infrastructure Securing



The Internet is Insecure!!

Did you know **BGP** has always been insecure?

- In 1997 One Autonomous System (AS) announced routes for most of the Internet.
- In 2008 Pakistan Telecom accidentally took down YouTube for much of the Internet.
- In 2010 A state-controlled China telecommunications company took 15% of the world's Internet traffic.
- In 2015 A broadband provider in India took out Google for most of the planet..

But there are solutions available! Secure your network today!

- The "**Resource Public Key Infrastructure**" (**RPKI**) allows operators to validate incoming routes
- Another security technology, "**BGPSEC**", fixes BGP on a hop-by-hop basis.
- And... **they work wonderfully together**



2 versus 4 Bytes





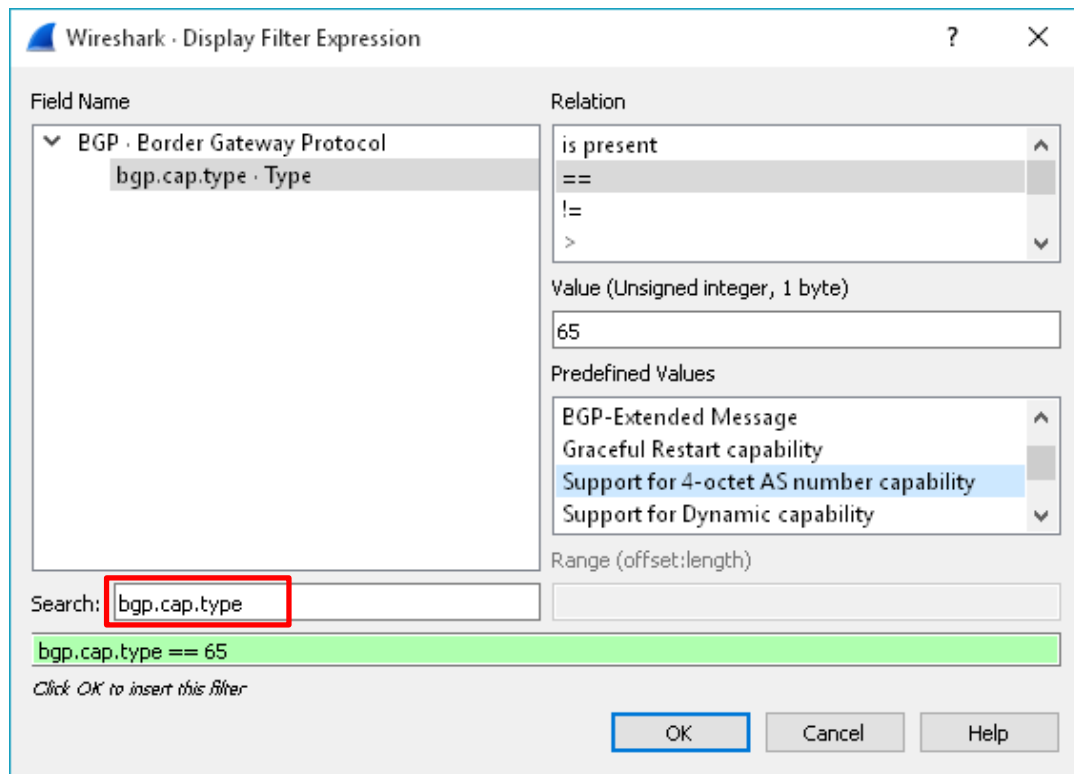
Four-Octet Autonomous System (AS) Number Space [RFC 6793]



- The Autonomous System number is encoded as a two-octet entity in the base BGP specification
- Exhaustion of the two-octet AS numbers
- „BGP carries the AS numbers in the "My Autonomous System" field of the OPEN message, in the AS_PATH attribute of the UPDATE message, and in the AGGREGATOR attribute of the UPDATE message. BGP also carries the AS numbers in the BGP Communities attribute.”
- Be aware of AS number 23456 (also called AS_TRANS)!



Four-Octet Autonomous System (AS) Number Space / DFE



- DFE is (your|one|my) way to learn or extend (our|one|many|my) protocol knowledge with Wireshark
- RFC reading is another way ;-)

HINT:
CTRL-C the Predefined Values for your notes



Four-Octet Autonomous System (AS) Number Space [RFC 6793]



Border Gateway Protocol - OPEN Message

Marker: ffffffffffffffffffffffffffffffff

Length: 105

Type: OPEN Message (1)

Version: 4

My AS: 64098

Hold Time: 9

BGP Identifier: ipt-transit-s1-ddos-loop.nsw.iptransit.com.au (59.153.11.4)

Optional Parameters Length: 76

Optional Parameters

> Optional Parameter: Capability

> Optional Parameter: Capability

> Optional Parameter: Capability

> Optional Parameter: Capability

v6multihop131b.pcap

> Optional Parameter: Capability

Parameter Type: Capability (2)

Parameter Length: 6

> Capability: Support for 4-octet AS number capability

Type: Support for 4-octet AS number capability (65)

Length: 4

AS Number: 64098

> Optional Parameter: Capability

> Optional Parameter: Capability

> Optional Parameter: Capability

> Optional Parameter: Capability

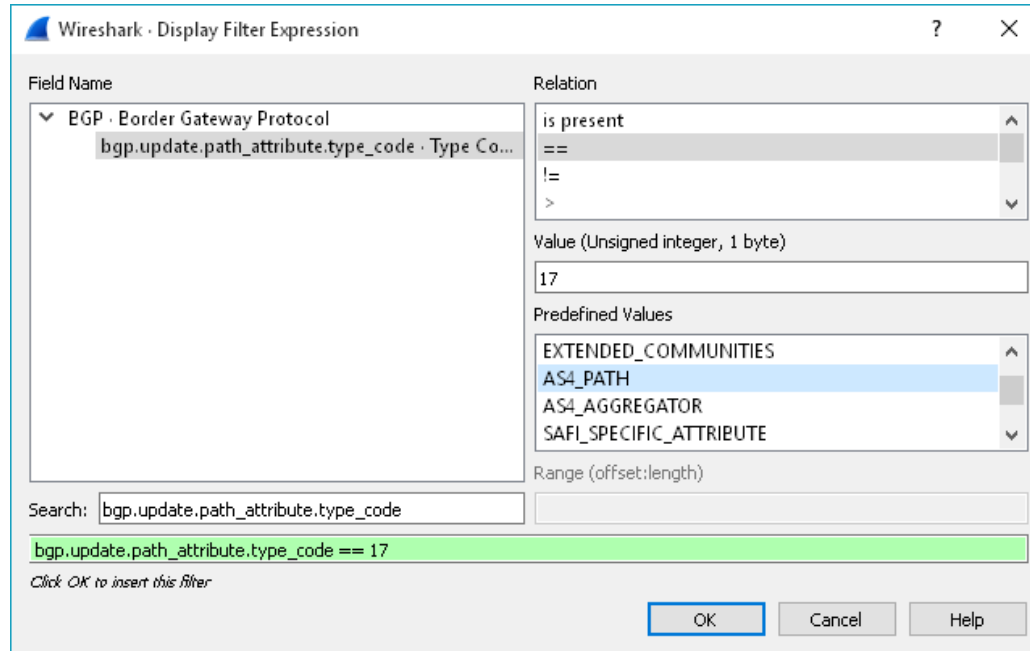
> Optional Parameter: Capability



Four-Octet Autonomous System (AS) Number Space [RFC 6793]



- AS4_PATH and AS4_AGGREGATOR





Wireshark Bug Database – Bug 10742

BGP: Incorrect decoding AS numbers when mixed AS size

[Wireshark Home](#) | [New](#) | [Browse](#) | [Search](#) | [Search](#) [\[?\]](#) | [Reports](#) | [Help](#) | [New](#)

Bug List: (48 of 85) [First](#) [Last](#) [Prev](#) [Next](#) [Show last search results](#)

[Bug 10742](#) - BGP: Incorrect decoding AS numbers when mixed AS size



Two/Four-Octet Interaction between BGP speakers



- AS_TRANS – AS 23456
- This AS number is also placed in the "My Autonomous System" field of the OPEN message originated by a NEW BGP speaker, if the speaker does not have a (globally unique) 2-octet AS number.

Border Gateway Protocol - OPEN Message

Marker: ffffffffffffffffffffffffffffffff
Length: 53
Type: OPEN Message (1)
Version: 4
My AS: 23456 (AS_TRANS)
Hold Time: 180
BGP Identifier: 192.168.12.2 (192.168.12.2)
Optional Parameters Length: 24

Optional Parameters

Parameter Type: Capability (2)
Parameter Length: 6

Capability: Support for 4-octet AS number capability
Type: Support for 4-octet AS number capability (65)
Length: 4
AS Number: 2222222

0000	45 c0 00 5d 52 df 40 00 01 06 8c a8 c0 a8 0c 02	E..]R.@.....
0010	c0 a8 0c 01 a8 dd 00 b3 8b c3 99 ea 27 d8 2d c1'.....
0020	50 18 40 00 73 50 00 00 ff ff ff ff ff ff ff ff	P.@.sP.....
0030	ff ff ff ff ff ff ff ff 00 35 01 04 5b a0 00 b45.[.....
0040	c0 a8 0c 02 18 02 06 01 04 00 01 00 01 02 02 80[.....
0050	00 02 02 02 00 02 06 41 04 01 53 15 8eA.S



Communities





Communities and extended ones



- Dissector reading

```
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965
6966
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993
6994
6995
6996
6997
6998
6999
7000
7001
7002
7003

case BGPTYPE_COMMUNITIES:
if (tlen % 4 != 0) {
    proto_tree_add_expert_format(subtree2, pinfo, &ei_bgp_length_invalid, tvb, o + i + aoff, tlen,
        "Communities (invalid): %u bytes", tlen,
        plurality(tlen, "", "s"));
    break;
}

proto_item_append_text(ti_pa, ": ");

ti_communities = proto_tree_add_item(subtree2, hf_bgp_update_path_attribute_communities,
    tvb, o + i + aoff, tlen);

communities_tree = proto_item_add_subtree(ti_communities, ET_SEQUENCE);

proto_item_append_text(ti_communities, "each community *");
/* (o + i + aoff) =
    (o + current attribute + aoff)
q = o + i + aoff;
end = q + tlen;

/* sniff each community */
while (q < end) {
    /* check for reserved values */
    uint32 community = tvb_get_uint32(tvb, q, 4);
    if ((community & 0xFFFF0000) == 0)
        proto_tree_add_item(communities_tree, hf_bgp_update_path_attribute_community_as,
            tvb, q, 4);
    else
        proto_item_append_text(ti_communities, "Reserved");
    community_tree = proto_item_add_subtree(communities_tree, ET_SEQUENCE);
    proto_tree_add_item(communities_tree, hf_bgp_update_path_attribute_community_as,
        tvb, q - 3 + aoff, 2, ENC_BIG_ENDIAN);
    proto_tree_add_item(communities_tree, hf_bgp_update_path_attribute_community_value,
        tvb, q - 1 + aoff, 2, ENC_BIG_ENDIAN);
    proto_item_append_text(ti_pa, "%u", tvb_get_ntohs(tvb, q - 3 + aoff));
    tvb_get_ntohs(tvb, q - 1 + aoff);
    proto_item_append_text(ti_communities, "%u", tvb_get_ntohs(tvb, q - 3 + aoff));
    tvb_get_ntohs(tvb, q - 1 + aoff);
    proto_item_append_text(ti_Community, ": %u", tvb_get_ntohs(tvb, q - 3 + aoff));
    tvb_get_ntohs(tvb, q - 1 + aoff);
}
q += 4;

break;
case BGPTYPE_ORIGINATOR_ID:
```

each community *
check for reserved values
uint32 community = tvb_get_uint32(tvb, q, 4);
if ((community & 0xFFFF0000) == 0)
proto_tree_add_item(communities_tree, hf_bgp_update_path_attribute_community_as, tvb, q, 4);
else
proto_item_append_text(ti_communities, "Reserved");
community_tree = proto_item_add_subtree(communities_tree, ET_SEQUENCE);
proto_tree_add_item(communities_tree, hf_bgp_update_path_attribute_community_as, tvb, q - 3 + aoff, 2, ENC_BIG_ENDIAN);
proto_tree_add_item(communities_tree, hf_bgp_update_path_attribute_community_value, tvb, q - 1 + aoff, 2, ENC_BIG_ENDIAN);
proto_item_append_text(ti_pa, "%u", tvb_get_ntohs(tvb, q - 3 + aoff));
tvb_get_ntohs(tvb, q - 1 + aoff);
proto_item_append_text(ti_communities, "%u", tvb_get_ntohs(tvb, q - 3 + aoff));
tvb_get_ntohs(tvb, q - 1 + aoff);
proto_item_append_text(ti_Community, ": %u", tvb_get_ntohs(tvb, q - 3 + aoff));
tvb_get_ntohs(tvb, q - 1 + aoff);



Communities and extended ones



- IANA assignments



```
1
2
3
4 Created 2005-08-15
5
6 Last Updated 2018-04-02
7
8
9 Available Formats
10 [IMG]
11 XML [IMG]
12 HTML [IMG]
13 Plain text
14
15 Registries included below
16
17 * BGP Transitive Extended Community Types
18 * BGP Non-Transitive Extended Community Types
19 * EVPN Extended Community Sub-Types
20 * Transitive Two-Octet AS-Specific Extended Community Sub-Types
21 * Non-Transitive Two-Octet AS-Specific Extended Community Sub-Types
22 * Transitive Four-Octet AS-Specific Extended Community Sub-Types
23 * Non-Transitive Four-Octet AS-Specific Extended Community Sub-Types
24 * Transitive IPv4-Address-Specific Extended Community Sub-Types
25 * Non-Transitive IPv4-Address-Specific Extended Community Sub-Types
26 * Transitive Opaque Extended Community Sub-Types
27 * Non-Transitive Opaque Extended Community Sub-Types
28 * Generic Transitive Experimental Use Extended Community Sub-Types
29 * Generic Transitive Experimental Use Extended Community Part 2 Sub-Types
30 * Generic Transitive Experimental Use Extended Community Part 3 Sub-Types
31 * Traffic Action Fields
32 * Transitive IPv6-Address-Specific Extended Community Types
33 * Non-Transitive IPv6-Address-Specific Extended Community Types
34 * Additional PMSI Tunnel Attribute Flags
35 * EVPN Layer 2 Attributes Control Flags
36 * E-Tree Flags
37 * Layer2 Info Extended Community Control Flags Bit Vector
38
```

Border Gateway Protocol (BGP) Extended Communities



- Extended Communities

- Carried extended communities: (1 community)
 - Route Target: 222:222 [Transitive 2-Octet AS-Specific]
 - Type: Transitive 2-Octet AS-Specific (0x00)
 - 0... = IANA Authority: Allocated on Standard Action, Early Allocation or Experimental Basis
 - .0.. = Transitive across AS: Transitive
 - Subtype (AS2): Route Target (0x02)
 - 2-Octet AS: 222
 - 4-Octet All: 222



Wireshark Bug Database – Bug 12631

BGP L2VPN EVPN Update with route type 2 incorrectly displayed as malformed

[Wireshark Home](#) | [New](#) | [Browse](#) | [Search](#) | | [Search](#) [?] | [Reports](#) | [Help](#) | [New Account](#) | [Log In](#) | [Forgot Password](#)

Bug List: (27 of 85) [First](#) [Last](#) [Prev](#) [Next](#) [Show last search results](#)

[Bug 12631](#) - BGP L2VPN EVPN Update with route type 2 incorrectly displayed as malformed

Status: RESOLVED FIXED



Large BGP Communities [RFC 8092]



```

▼ Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffffffff
  Length: 75
  Type: UPDATE Message (2)
  Withdrawn Routes Length: 0
  Total Path Attribute Length: 47
  ▼ Path attributes
    > Path Attribute - ORIGIN: IGP
    > Path Attribute - AS_PATH: 65536
    > Path Attribute - NEXT_HOP: 192.0.2.2
    ▼ Path Attribute - LARGE_COMMUNITY: 65535:1:1 4294967295:4294967295:4294967295
      > Flags: 0xc0, Optional, Transitive, Complete
      Type Code: LARGE_COMMUNITY (32)
      Length: 24
      ▼ Large communities: 65535:1:1
        Global Administrator: 65535
        Local Data Part 1: 1
        Local Data Part 2: 1
      ▼ Large communities: 4294967295:4294967295:4294967295
        Global Administrator: 4294967295
        Local Data Part 1: 4294967295
        Local Data Part 2: 4294967295

```



BGP Load Balance



https://github.com/Juniper/contrail-controller/blob/master/src/bgp/extended-community/load_balance.h#L24

```

20  /*
21  * BGP LoadBalance Opaque Extended Community with SubType 0xAA (TBA)
22  *
23  * 0          1          2          3
24  * 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
25  * +-----+-----+-----+-----+-----+-----+-----+-----+
26  * | Type  0x03   | Sub-Type 0xAA |s d c p P R R R|R R R R R R R R|
27  * +-----+-----+-----+-----+-----+-----+-----+-----+
28  * | Reserved    |B R R R R R R R| Reserved    | Reserved    |
29  * +-----+-----+-----+-----+-----+-----+-----+-----+
30  *
31  * Type: 0x03 Opaque
32  * SubType: 0xAA LoadBalance attribute information (TBA)
33  * s: Use l3_source_address ECMP Load-balancing
34  * d: Use l3_destination_address ECMP Load-balancing
35  * c: Use l4_protocol ECMP Load-balancing
36  * p: Use l4_source_port ECMP Load-balancing
37  * P: Use l4_destination_port ECMP Load-balancing
38  * B: Use source_bias (instead of ECMP load-balancing)
39  * R: Reserved
40  */

```



0xFFFF029A



- BGP blackhole filtering is a routing technique used to drop unwanted traffic
- Which value is this – 0x29A → 666 !!!
- Reading the RFCs again and use your favorite search engine

→ 65535:666 = 0xFFFF029A is from the well--known BGP community space



BGP-LS





BGP protocol with Link-State Distribution



- Link-State (IGP – OSPFv2/v3 or ISIS) Distribution Using BGP
- Use case are SDNs, where the network can be programmatically controlled by a centralized controller
- BGP-LS becomes important when LSP paths cross multiple routing domains or when IGP routing information is required by external entities such as ALTO or PCE servers for optimized path computation
- <https://wiki.onosproject.org/display/ONOS/BGP+protocol+with+Link-State+Distribution>



BGP Link-State extensions for Segment Routing



- BGP Link-State (BGP-LS) is an Address Family Identifier (AFI) and Sub-address Family Identifier (SAFI) defined to carry interior gateway Protocol (IGP) link-state database through BGP
- In order to address the need for applications that require topological visibility across IGP areas, or even across Autonomous Systems (AS), the BGP-LS address-family/sub-address-family have been defined to allow BGP to carry Link-State information. The BGP Network Layer Reachability Information (NLRI) encoding format for BGP-LS and a new BGP Path Attribute called the BGP-LS attribute are defined in [RFC 7752]



BGP Link-State in packet-bgp.c



```
2050  /* FF: BGP-LS is just a collector of IGP link state information. Some
2051     fields are encoded "as-is" from the IGP, hence in order to dissect
2052     them properly we must be aware of their origin, e.g. IS-IS or OSPF.
2053     So, *before* dissecting LINK_STATE attributes we must get the
2054     'Protocol-ID' field that is present in the MP_[UN]REACH_NLRI
2055     attribute. The tricky thing is that there is no strict order
2056     for path attributes on the wire, hence we have to keep track
2057     of 1) the 'Protocol-ID' from the MP_[UN]REACH_NLRI and 2)
2058     the offset/len of the LINK_STATE attribute. We store them in
2059     per-packet proto_data and once we got both we are ready for the
2060     LINK_STATE attribute dissection.
2061  */
```



BGP Link-State



```

  ▾ Border Gateway Protocol - OPEN Message
    Marker: ffffffffffffffffffffffffffffffffff
    Length: 61
    Type: OPEN Message (1)
    Version: 4
    My AS: 65100
    Hold Time: 180
    BGP Identifier: 163.162.95.53 (163.162.95.53)
    Optional Parameters Length: 32
  ▾ Optional Parameters
    ▾ Optional Parameter: Capability
      Parameter Type: Capability (2)
      Parameter Length: 6
    ▾ Capability: Multiprotocol extensions capability
      Type: Multiprotocol extensions capability (1)
      Length: 4
      AFI: BGP-LS (16388)
      Reserved: 00
      SAFI: BGP-LS (71)

```




BGP Link-State NLRI



- ▼ Path attributes
 - ▼ Path Attribute - MP_REACH_NLRI
 - > Flags: 0x90, Optional, Extended-Length, Non-transitive, Complete
 - Type Code: MP_REACH_NLRI (14)
 - Length: 3852
 - Address family identifier (AFI): BGP-LS (16388)
 - Subsequent address family identifier (SAFI): BGP-LS (71)
 - ▼ Next hop network address (4 bytes)
 - Next Hop: 10.0.0.208
 - Number of Subnetwork points of attachment (SNPA): 0
 - ▼ Network layer reachability information (3843 bytes)
 - ▼ BGP-LS NLRI
 - NLRI Type: IPv4 Topology Prefix NLRI (3)
 - NLRI Length: 59
 - ▼ Link-State NLRI IPv4 Topology Prefix
 - Protocol ID: OSPF (3)
 - Identifier: Unknown (2)
 - > Local Node Descriptors TLV
 - > Prefix Descriptors TLV
- ▼ Link-State NLRI IPv4 Topology Prefix
 - Protocol ID: OSPF (3)
 - Identifier: Unknown (2)
 - ▼ Local Node Descriptors TLV
 - Type: 256
 - Length: 32
 - ▼ Autonomous System TLV
 - Type: 512
 - Length: 4
 - AS ID: 65060 (0x0000fe24)
 - ▼ BGP-LS Identifier TLV
 - Type: 513
 - Length: 4
 - BGP-LS ID: 167772368 (0x0a0000d0)
 - ▼ Area ID TLV
 - Type: 514
 - Length: 4
 - Area ID: 758001410 (0x2d2e2f02)
 - ▼ IGP Router-ID
 - Type: 515
 - Length: 4
 - IGP ID: 0a0000d0
 - > Prefix Descriptors TLV



BGP-LS Path Attributes



- The BGP-LS attribute is an optional, non-transitive BGP attribute that is used to carry link, node, and prefix parameters and attributes
- These Path attributes are categorized into two categories:
 - Node Attributes with TLVs
 - Link Attributes with TLVs



BGP Link-State in bugs.wireshark.org



- https://bugs.wireshark.org/bugzilla/show_bug.cgi?id=13841
- https://bugs.wireshark.org/bugzilla/show_bug.cgi?id=12060





BGP Additional Path





BGP Additional Path



BGP Additional Path | line 1017 | packet-bgp.c

```
/*  
 * Detect IPv4 prefixes conform to BGP Additional Path but NOT conform to standard BGP  
 *  
 * A real BGP speaker would rely on the BGP Additional Path in the BGP Open messages.  
 * But it is not suitable for a packet analyse because the BGP sessions are not supposed to  
 * restart very often, and Open messages from both sides of the session would be needed  
 * to determine the result of the capability negotiation.  
 * Code inspired from the decode_prefix4 function  
 */
```



BGP Additional Path



Optional Parameter

- ▼ Optional Parameter: Capability
 - Parameter Type: Capability (2)
 - Parameter Length: 10
 - ▼ Capability: Support for Additional Paths
 - Type: Support for Additional Paths (69)
 - Length: 8
 - AFI: IPv4 (1)
 - SAFI: Unicast (1)
 - Send/Receive: Receive (1)
 - AFI: Layer-2 VPII (25)
 - SAFI: EVPPII (70)
 - Send/Receive: Receive (1)

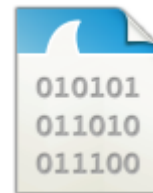


BGP Errors



BGP session flaps with add path enabled

final.pcap



<https://jira.opendaylight.org/secure/attachment/13230/final.pcap>



BGP EVPN





BGP and EVPN



- BGP EVPN (Ethernet Virtual Private Network) relies on basic BGP and MP-BGP extensions
- The extensions can carry reachability information (NLRI) for multiple protocols (especially EVPN)
- EVPN is a technology that is used to extend Ethernet circuits across Data Center, Data Center Interconnect (DCI) and Service Provider networks
- Treat MAC addresses and distribute them via BGP
- It is expected to succeed other L2VPN transport methods such as BGP-based L2VPN [RFC 6624]
- EVPN is technically just another address family in Multi Protocol (MP) BGP [RFC 7432] - MAC Mobility extended community is defined there ;-)



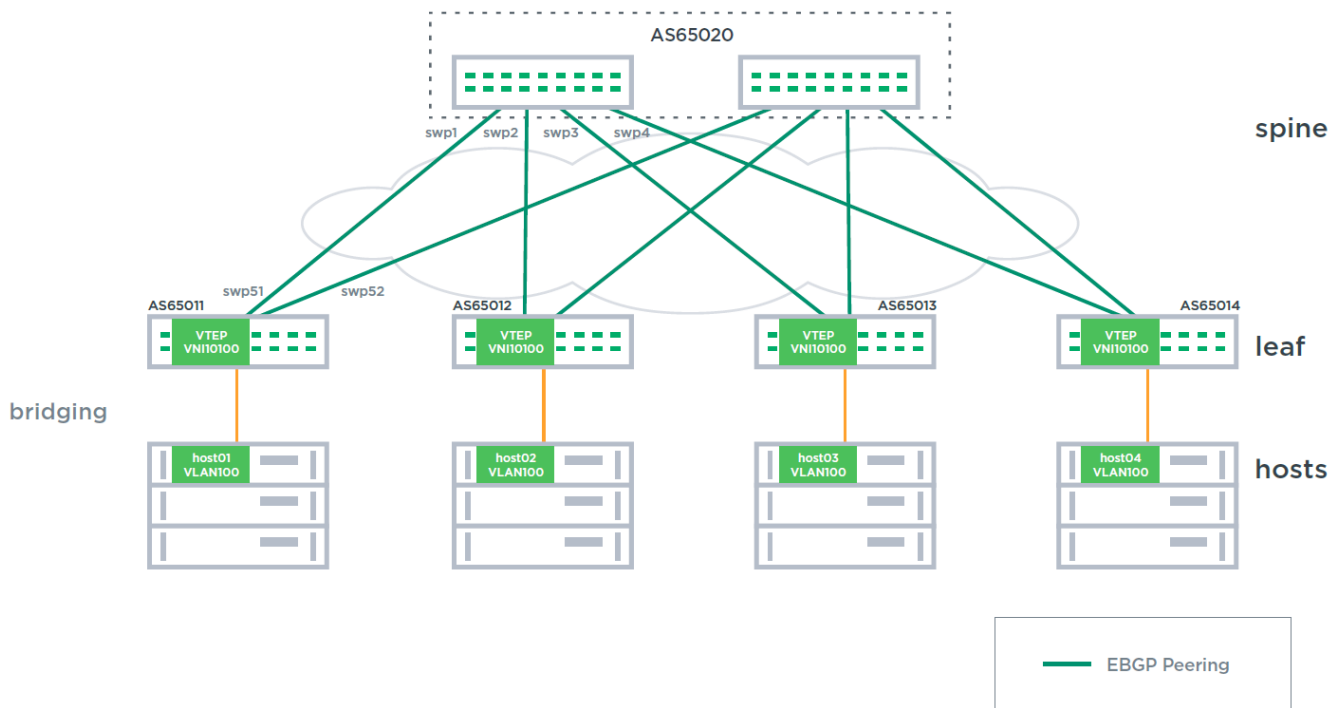
EVPN and RFC 7432



- EVPN address family carries both Layer 2 and Layer 3 reachability information. This provides integrated bridging and routing in overlay networks
- RFC 7432 defines different route types:
 - 0 Reserved
 - 1 Ethernet Auto-discovery
 - 2 MAC/IP Advertisement
 - 3 Inclusive Multicast Ethernet Tag
 - 4 Ethernet Segment
 - ...
- Enables traffic load balancing for multihomed CEs with ECMP MAC routes



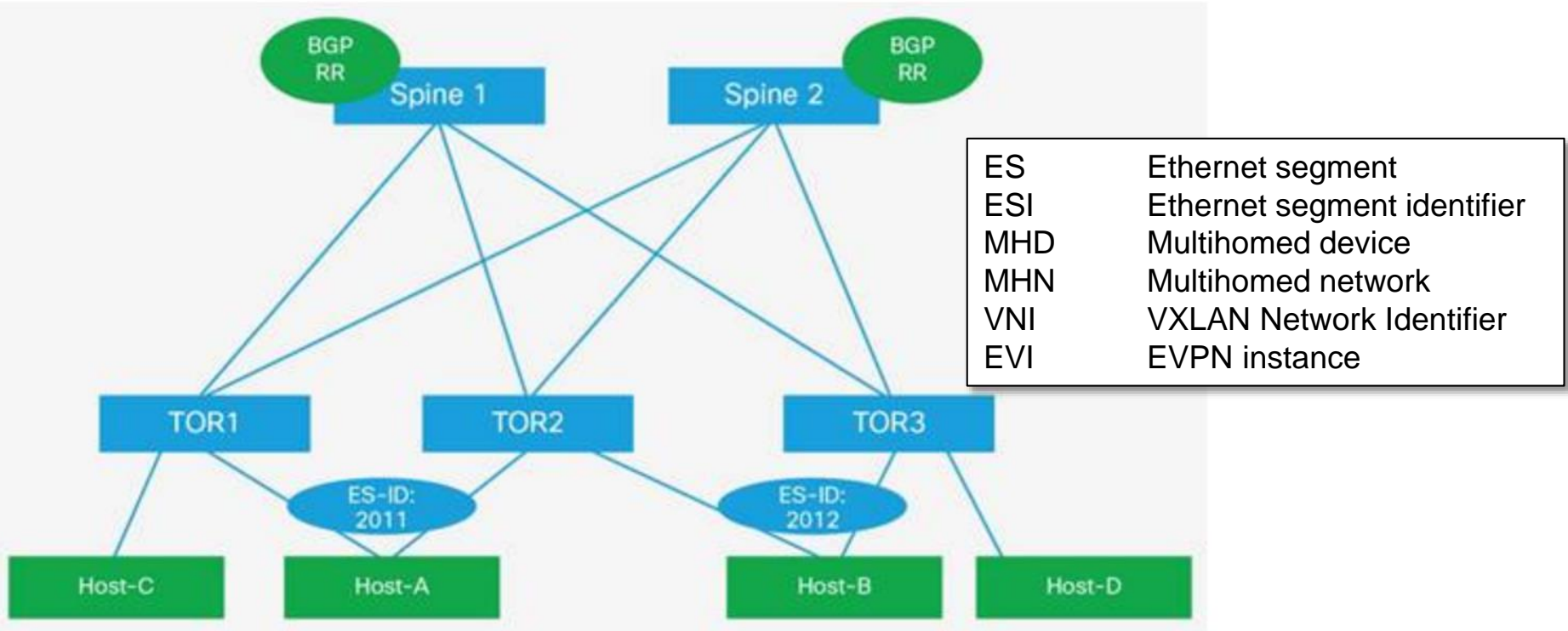
BGP and EVPN



Source: <https://cumulusnetworks.com/>

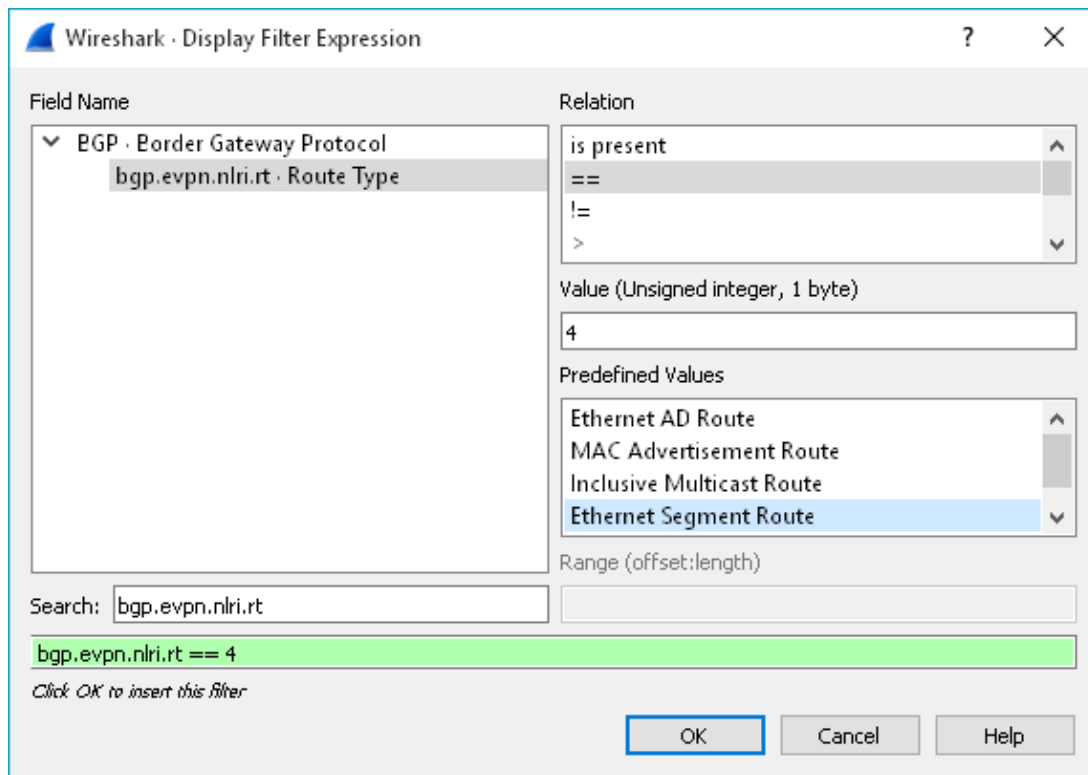


BGP, EVPN, VXLAN and Multihoming

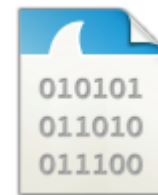




BGP EVPN Route types



- `bgp.evpn.nlri.rt`



- `cisco-bgp.pcap`
- `EVPN Route Types BGP Capture.pcap`



BGP Graceful Restart





Graceful Restart Mechanism for BGP



- Graceful Restart Capability – RFC 4724
- Long-Lived Graceful Restart (LLGR) Capability – draft
- Let the partner know if the session is from a restart
- how long to wait before dropping stale routes
- Per AFI/SAFI !



v6multihop131b.pcap

```
Border Gateway Protocol - OPEN Message
Marker: ffffffffffffffffffffffffffffffff
Length: 105
Type: OPEN Message (1)
Version: 4
My AS: 64098
Hold Time: 9
BGP Identifier: 59.153.11.4 (59.153.11.4)
Optional Parameters Length: 76
Optional Parameters
  Optional Parameter: Capability
  Optional Parameter: Capability
  Optional Parameter: Capability
  Optional Parameter: Capability
  Optional Parameter: Capability
  Optional Parameter: Capability
  Optional Parameter: Capability
  Optional Parameter: Capability
  Optional Parameter: Capability
  Optional Parameter: Capability
  Parameter Type: Capability (2)
  Parameter Length: 8
  Capability: Graceful Restart capability
    Type: Graceful Restart capability (64)
    Length: 6
    Restart Timers: 0x0078
      0... .. = Restart: No
      ... 0000 0111 1000 = Time: 120
    AFI: IPv6 (2)
    SAFI: Unicast (1)
  Flag: 0x00
    0... .. = Preserve forwarding state: No
```




https://bugs.wireshark.org/bugzilla/show_bug.cgi?id=7734



Wireshark Bug Database – Bug 7734

BGP bad decoding for Graceful Restart Capability with only helper support & for Enhanced Route Refresh Capability

<ul style="list-style-type: none">Optional Parameter: Capability<ul style="list-style-type: none">Parameter Type: Capability (2)Parameter Length: 4Capability: Graceful Restart capability<ul style="list-style-type: none">Type: Graceful Restart capability (64)Length: 2[Expert Info (Chat/Request): Graceful Restart Capability sRestart Timers: 0x8078, Restart<ul style="list-style-type: none">1... .. = Restart: Yes.... 0000 0111 1000 = Time: 120		<ul style="list-style-type: none">Optional Parameter: Capability<ul style="list-style-type: none">Parameter Type: Capability (2)Parameter Length: 8Capability: Graceful Restart capability<ul style="list-style-type: none">Type: Graceful Restart capability (64)Length: 6Restart Timers: 0x0078<ul style="list-style-type: none">0... .. = Restart: No.... 0000 0111 1000 = Time: 120AFI: IPv6 (2)SAFI: Unicast (1)Flag: 0x00<ul style="list-style-type: none">0... .. = Preserve forwarding state: No
--	--	---



BGP FlowSpec NLRI in bugs.wireshark.org



Wireshark Bug Database – Bug 12568

Wireshark is marking BGP FlowSpec NLRI as malformed if NLRI length is larger than 239 bytes

[Wireshark Home](#) | [New](#) | [Browse](#) | [Search](#) | [Search](#) [\[?\]](#) | [Reports](#) | [Help](#) | [New Account](#) | [Log In](#) | [Forgot Password](#)

[Bug 12568](#) - Wireshark is marking BGP FlowSpec NLRI as malformed if NLRI length is larger than 239 bytes



Wireshark Bug Database – Bug 8691

Adding support of BGP flow spec RFC 5575

[Wireshark Home](#) | [New](#) | [Browse](#) | [Search](#) | [Search](#) [\[?\]](#) | [Reports](#) | [Help](#) | [New Account](#) | [Log In](#) | [Forgot Password](#)

[Bug 8691](#) - Adding support of BGP flow spec RFC 5575



Maybe next steps?



BGP over HTTP/2 with QUIC



Ilari Stenroth  @istenrot · 16. Apr.

When we'll get **#BGP** over HTTP/2? In the end should run every protocol over HTTP/2! Just imagine all cool **#IoT** applications for BGP over HTTP/2. **#sarcasm**





Useful implementation

Playing battleships over BGP



```
her side played a E4
Your Side
|A|B|C|D|E|F|G|H|I|J|
0| | | | | | | | | | |0
1| | | | | | | | | | |1
2| | | | | | | | | | |2
3| | | | | | | | | | |3
4| | | | | | | | | | |4
5| | | | | | | | | | |5
6| | | | | | | | | | |6
7| | | | | | | | | | |7
8| | | | | | | | | | |8
9| | | | | | | | | | |9
|A|B|C|D|E|F|G|H|I|J|
[000001] Next Move> F3
Firing on F3
```

```
Player Two
|A|B|C|D|E|F|G|H|I|J|
0| | | | | | | | | | |0
1| | | | | | | | | | |1
2| | | | | | | | | | |2
3| | | | | | | | | | |3
4| | | | | | | | | | |4
5| | | | | | | | | | |5
6| | | | | | | | | | |6
7| | | | | | | | | | |7
8| | | | | | | | | | |8
9| | | | | | | | | | |9
|A|B|C|D|E|F|G|H|I|J|
```



Please provide Session Feedback



#sf18eu • Imperial Riding School Renaissance Vienna • Oct 29 - Nov 2



Thank you!



- In secret service since 1999
- Will conquer the world in 2018
 - Yes, really!
 - What? You don't believe me?

